Preparing Data for Analysis Using R:

Basic through Advanced Techniques
Part 2: statistical issues

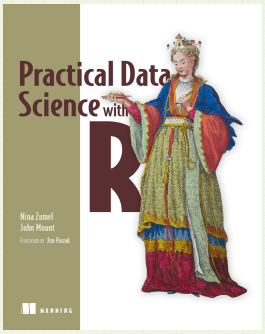
John Mount & Nina Zumel
Win-Vector, LLC

All materials: https://github.com/WinVector/PreparingDataWorkshop

1

Who I am

- John Mount
- Principal Consultant at Win-Vector LLC
- One of the authors of Practical Data Science with R



2

Statistical issues

- Up until now we have largely been working around real-world *operational* issues
  - Missing values
  - Novel levels
  - Categorical variables with very many levels
- Even more dangerous are statistical issues, both those obvious and those unnoticed

3

Statistical issues we are worried about

- Bias
  - Systematic mis-predictions that are function more of the fitting process than the data.
- Overfitting
  - Models that perform well on training data and then fail in production.

4

Why we care

- For "wide data" (very many variables) we can not safely leave all variable selection to common machine learning software.
  - Huge multiple comparison problem
- We can accidentally introduce one issue ourselves when treating variables.
  - Need to at least fix our own mistakes.

5

Is Variable Selection a Data Treatment Problem?

- For very wide data sets: yes!
  - Too many variables slow down model fitting
  - Can result in misleading models
- So should at least prune variables with no obvious signal

6

Machine Learning procedures assume curated variables

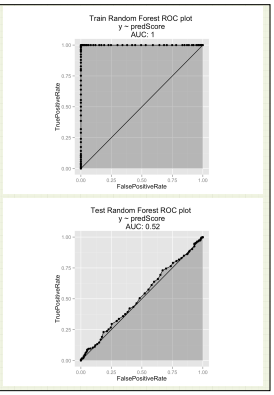
- For *at least* the following common popular machine learning algorithms we can design a simple data set where we get arbitrarily high accuracy on training even when the dependent variable is generated completely independently of all of the independent variables.
  - Decision Trees
  - Logistic Regression
  - Elastic Net Logistic Regression
  - Gradient Boosting
  - Naive Bayes
  - Random Forest
  - Support Vector Machine

7

7

Can't we keep at least some of our training performance?

- Common situation:
  - Near perfect fit on training data.
  - Model performs like random guessing on new instances.
  - Extreme over fit.
- One often hopes some regularized, ensemble, or transformed version of such a model would have at least some use on new instances.



8

8

Work through BadModels BadModel example

9

What causes this?

- "Regression to the mean"
  - Some fraction of your "top performers" performance is actually noise.
  - This "luck" isn't repeated later in production.
  - With enough variables to choose from, your top performers can be completely due to noise (and cut in front of all true variables).
- Also called "Freedman's paradox"
  - Freedman, D. A. (1983) "A note on screening regression equations." The American Statistician, 37, 152–155.

10

Why doesn't bagging or regularization always fix this?

- Bagging can help eliminate modeling variance, this is bias.
  - So we make similar mistakes in each sub-model.
  - Duplicate or near duplicate variables defeat sampling based sub-model diversity (such as Random Forest's variable controls.
- Regularization (or controlling model complexity) can help
  - Most regularization ends up looking like some sort of prior
  - Often defeated when you have a lot of data (Bernstein–von Mises theorem) and when you have a lot of multiple comparison bias (many models/variables to pick from).

11

What to do

- Variable selection by significance
  - Pick significance 1/"number of proposed variables"
    - With this significance only a few completely useless variables should leak into the model
    - Downstream model system should be able to deal with these
  - Notice significance pick is different than "always use 0.05" nonsense
- We will justify this later

12

Work through BadModels yAwareReduction example

13

Are we done?

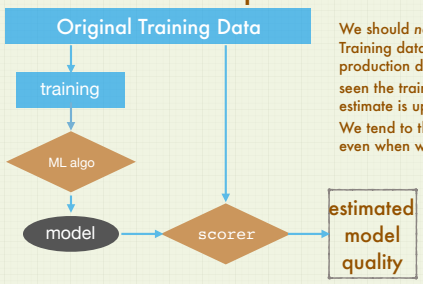
14

"We can accidentally introduce one issue ourselves when treating variables"

- Impact/effect coding is *not* completely safe
  - An impact/effect coded categorical is essentially a model
  - So any model using such variables is a nested model
  - Nested models require some additional care

15

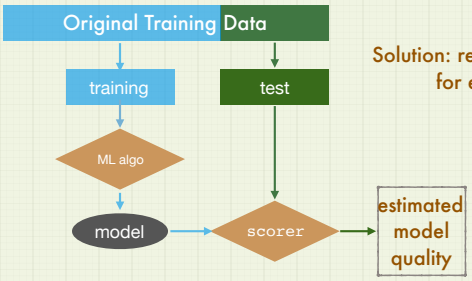
Naive machine learning practice



We should never do this.
Training data isn't exchangeable with future production data (due to the trainer having seen the training data), so the model quality estimate is upwardly biased.
We tend to think we have a good model, even when we do not.

16

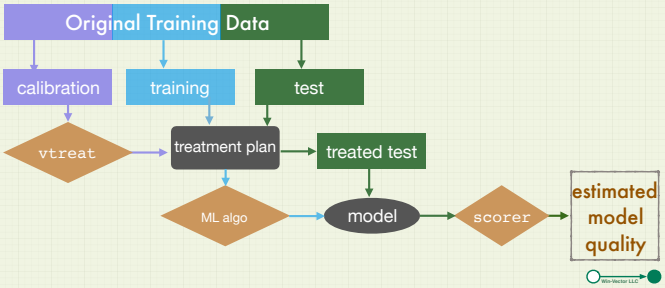
Standard machine learning practice



Solution: reserve some data for evaluation

17

One solution: use a calibration set to fit impact models



18

Why so much separation?

- vtreat is essentially building models for the large categorical columns.
- If data that was used in designing the treatment plan is used in training- the training system tends to think these variables are way more reliable than they actually are, and vastly under estimate the number of degrees of freedom such variables consume.

19

Run nested examples

20

Why does significance pruning work?

- Essentially it imitates a really neat permutation test experiment.

21

Thought Experiment: What does No Signal look like?

Data set: y depends on x

x	y
x1	y1
x2	y2
x3	y3
x4	y4
…	…
xn	yn

Permute y: now y has no relation to x

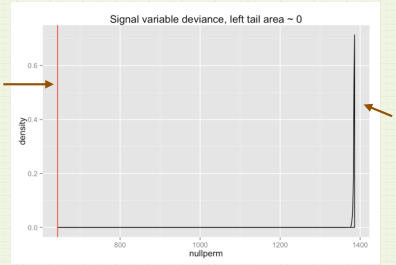
x	y
x1	y4
x2	y10
x3	y7
x4	y1
…	…
xn	yk

Do this several times…

22

Fit models, and compare: When y depends on x

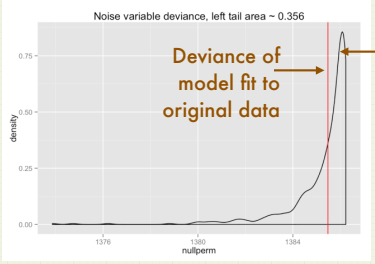
Deviance of model fit to original data



Distribution of deviances of models fit to permuted data

23

When y has no relation to x



Deviance of model fit to original data

Distribution of deviances of models fit to permuted data

Left tail area: The significance of the model fit to original data

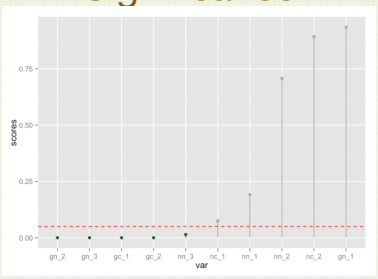
24

To test a variable v for signal

- Ideally: Build a one-variable model on v, check its significance by a permutation test.
- In practice: use a chi-squared or F-test
  - Model significance of logistic or linear regression, respectively

25

Filtering by Variable Significance



26

Assumptions

- A variable with signal will manifest it through a linear model or a Bayes model.
- We are not considering interactions among variables when looking for signal
  - That is the left to the machine learning modeling

27

Conclusions

- You must prepare data prior to analysis, even when using sophisticated modern machine learning methods.
- Even though you may be preparing your data for mere operational reasons (data cleaning), you soon run into statistical issues.
- Think of columns and variables as single variable models.
- There are many good techniques to correctly and efficiently build sub-models.

28

Further Reading

- vtreat
  - https://cran.r-project.org/package=vtreat
- Model testing procedures
  - http://www.win-vector.com/blog/2015/09/isyourmodelgoingtowork/
- Permutation tests
  - http://www.win-vector.com/blog/2015/08/how-do-you-know-if-your-data-has-signal/
- Differential privacy
  - http://www.win-vector.com/blog/2015/11/our-differential-privacy-mini-series/

29

Thank You

All materials: https://github.com/WinVector/PreparingDataWorkshop

30